24.3.2 Klasse MimePart

Die Klasse MimePart der Komponente *gb.mime* repräsentiert einen Teil im Text-Körper (Body) einer MimeMessage → Kapitel 24.3.3 Klasse MimeMessage. Eine Leerzeile trennt den Header einer EMail formal vom Body. Diese Klasse können Sie erzeugen.

24.3.2.1 Eigenschaften

Die Klasse *MimePart* hat folgende Eigenschaften:

Eigenschaft	Datentyp	Beschreibung
Count	Integer	Gibt die Anzahl der Teile einer MimeMessage zurück.
ContentDisposition	String	Setzt das Header-Feld 'Content-Disposition' einer MimeMessage oder gibt dessen Wert zurück.
ContentEncoding	Integer	Gibt die Art der Kodierung eines Teils in einer MimeMessage zurück (7Bit, 8Bit, Base64, Binary, Default, QuotedPrintable, UUEncoding.
ContentType	String	Gibt den Wert des Header-Feldes 'Content-Typ' einer MimeMessage zurück.
ContentId	String	Setzt das Header-Feld 'Content-ID' einer MimeMessage oder gibt dessen Wert zurück.
Disposition	String	Setzt das Header-Feld 'Disposition' einer MimeMessage oder gibt dessen Wert zurück.
FileName	String	Gibt den Datei-Namen des Anhangs zurück, der im Header-Feld 'Disposition' einer MimeMessage festgelegt wurde. Wird NULL zurückgegeben, dann wurde keine Datei festgelegt.
Message	MimeMessage	Setzt den Message-Teil innerhalb einer MimePart-Struktur oder gibt diesen Message-Teil zurück.
Data	String	Gibt die *automatisch* dekodierten Nutzer-Daten eines Teils einer MimeMessage zurück. Achten Sie darauf, den Typ der Daten zu überprüfen.
Headers	MimePart.Headers	*MimePart.Headers[fieldname]* gibt die spezifizierten Header-Felder eines Teils zurück.

Tabelle 24.3.2.1.1 : Eigenschaften der Klasse MimePart

Beispiel: Quelltext-Ausschnitte einer EMail vom Typ MimeMessage

```
[1]  ...
[2]  Return-path: <xxxx.xxx@freenet.de>
[3]  ...
[4]  To: yyyy@freenet.de
[5]  From: YYYY <xxxx.xxx@freenet.de>
[6]  Subject: Bild-Datei
[7]  Message-ID: <56828856.4070805@freenet.de>
[8]  Date: Tue, 29 Dec 2015 14:19:18 +0100
[9]  ...
[10] MIME-Version: 1.0
[11] Content-Type: multipart/mixed; boundary="000100040309080505050507"
[12] ...
[13] <LEERZEILE TRENNT HEADER UND BODY>
[14] This is a multi-part message in MIME format.
[15] --000100040309080505050507
[16] Content-Type: multipart/alternative; boundary="030406090702040408090509"
[17]
[18] --030406090702040408090509
[19] Content-Type: text/plain; charset=utf-8; format=flowed
[20] Content-Transfer-Encoding: 7bit
[21]
[22] Hallo Hans,
[23] im Anhang die angeforderte Bild-Datei '8.png'.
[24] Mit freundlichem Gruss
[25] Honsek
[26]
[27] --030406090702040408090509
[28] Content-Type: text/html; charset=utf-8
[29] Content-Transfer-Encoding: 7bit
[30]
```

```
[31] <html>
[32]   <head>
[33]     <meta http-equiv="content-type" content="text/html; charset=utf-8">
[34]   </head>
[35]   <body text="#000000" bgcolor="#FFFFFF">
[36]     <font size="+1">Hallo Hans,<br>
[37]       im Anhang die angeforderte Bild-Datei '8.png'.<br>
[38]       Mit freundlichem Gruss<br>
[39]       Honsek<br>
[40]     </font>
[41]   </body>
[42] </html>
[43]
[44] --030406090702040408090509--
[45]
[46] --000100040309080505050507
[47] Content-Type: image/png; name="8.png"
[48] Content-Transfer-Encoding: base64
[49] Content-Disposition: attachment; filename="8.png"
[50]
[51] iVBORw0KGgoAAAANSUhEUgAAAAgAAAAICAYAAADED76LAAAAO0lEQVQYlX2PyQ0AMAjDnKr7
[52] r5x+egvIEyxwZDBFOoASxJoAEWUB0J6Zhedi5QFSh/0lkDnAd/pWK2sORyMOEhaL7BAAAAAA
[53] SUVORK5CYII=
[54]
[55] --000100040309080505050507--
```

Struktur:

```
+ multipart/mixed
|
+-+ multipart/alternative  2
| |
| +-- text/plain  0
| |
| +-- text/html  0
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
ANHANG 1 :  8.png attachment image/png
```

Abbildung 24.3.2.1.1: Darstellung 1 der Struktur der o.a. MimeMessage

Ein Parser gibt die Struktur einer MimeMessage aus. Alle Teile der MimeMessage und ihre Hierarchie werden symbolisch darstellt.

- Sie sehen in der Abbildung 24.3.2.1.1 die Struktur eines EMail-Quelltextes, der im Beispiel aus mehreren Teilen (Typ: multipart/mixed) besteht – einem Text-Teil und einem Anhang-Teil.
- Der Text-Teil besteht aus zwei Texten (Typ: multipart/alternative) mit gleichem Inhalt, jedoch mit unterschiedlichen, alternativen Formaten. Der erste Text ist vom Typ text/plain und der zweite ist vom Typ text/html. Die Text-Teile sind nicht weiter verschachtelt.
- Im Anhang liegt das Bild 8.png vom Typ image/png.

Kommentar:

- In den Zeilen 1 bis 55 sehen Sie einen Ausschnitt aus dem Quelltext einer EMail. Im Gegensatz zur darunter angezeigten Struktur sehen Sie den konkreten Inhalt.
- Die Abgrenzungen der einzelnen Teile – die sogenannten *boundary* – sind farbig hervorgehoben. Jede neue Abgrenzung beginnt mit zwei Bindestrichen, denen der *Wert* der Abgrenzung folgt. Das Ende eines Teils wird durch --WertDerAbgrenzung-- markiert. Nach jeder neuen Abgrenzung folgt ein eigener (Sub-)Header mit unterschiedlichen Header-Feldern. Ein Beispiel dafür zeigen die Zeile 46 sowie die Zeilen 47 bis 49.

24.3.2.2 Methoden

Die Klasse *MimePart* verfügt nur über zwei Methoden:

Methode	Beschreibung
Add (Part As MimePart)	Fügt 'Part' als Teil der aktuellen MimeMessage hinzu.
ToString ( )	Die Funktion fügt alle mit der Add-Methode erzeugten Teile zu einer MimeMessage zusammen.

Tabelle 24.3.2.2.1 : Methoden der Klasse MimePart

Eine Beschreibung des Einsatzes der beiden Methoden finden Sie in einem Projekt im Kapitel 24.4.3.

24.3.3 Beispiel Struktur-Parser

Der vorgestellte Parser erzeugt eine Sicht auf die Struktur einer MimeMessage. Untersucht werden dabei der Header und die einzelnen Teile im Body einer MimeMessage:

Abbildung 24.3.3.1: Darstellung 2 der Struktur einer weiteren MimeMessage

Der Quelltext für den Parser wird vollständig angegeben:

```
[1]  ' Gambas class file
[2]
[3]  Private $iLevel As Integer
[4]  Private $iFlag As Integer
[5]  Private $sBodyType As String
[6]  Private $iAttachmentCount As Integer
[7]
[8]  Public Sub Form_Open()
[9]    FMain.Center
[10]   FMain.Utility = True
[11]   btnGetStructureMimeMessage.Enabled = False
[12] End ' Form_Open()
[13]
[14] Public Sub btnClose_Click()
[15]   FMain.Close
[16] End ' btnClose_Click()
[17]
[18] Public Sub btnGetStructureMimeMessage_Click()
[19]   Dim hMimeMessage As New MimeMessage
[20]   Dim sMessage As String
[21]
[22]   hMimeMessage = New MimeMessage(txaMonitor.Text)
[23]
[24]   $sBodyType = Scan(hMimeMessage.Part.Headers["Content-Type"], "*;*")[0]
[25]
[26]   sMessage = ("S T R U C T U R E  M I M E - M E S S A G E")
[27]   txaMonitor.Insert(sMessage & gb.NewLine)
[28]   txaMonitor.Insert(String$(String.Len(sMessage), "-") & gb.NewLine)
[29]   txaMonitor.Insert(gb.NewLine)
[30]
[31] ' Parser
[32]   txaMonitor.Insert("+ " & $sBodyType & gb.NewLine)
[33]   ParsePart(hMimeMessage.Body, True)
[34]   ParsePart(hMimeMessage.Part, False)
[35]
[36]   btnGetStructureMimeMessage.Enabled = False
[37]
[38] End ' btnGetStructureMimeMessage_Click()
[39]
[40] Public Sub ParsePart(hPart As MimePart, IsBody As Boolean)
[41]    Dim hChild As MimePart
[42]    Dim sMessage As String
[43]
[44]    If IsBody = True Then
[45]       txaMonitor.Insert(String$($iLevel, "| ") & "| " & gb.NewLine)
[46]       txaMonitor.Insert(String$($iLevel, "| ") & "+-" & If(hPart.Count, "+ ", "- "))
[47]       sMessage = hPart.ContentType & " " & hPart.FileName
[48]       sMessage &= " " & hPart.Count & gb.NewLine
[49]       txaMonitor.Insert(sMessage)
[50]    Else
```

```
[51]       If hPart.Disposition = "attachment" And Str(hPart.Count) = 0 Then
[52]          If $iFlag = 0 Then
[53]             txaMonitor.Insert("|" & gb.NewLine)
[54]             Inc $iFlag
[55]          Endif
[56]          sMessage = ("+ Attachment ") & Str($iAttachmentCount + 1) & ": " & " " & hPart.FileName
[57]          sMessage &= " " & hPart.Disposition & " " & hPart.ContentType & gb.NewLine
[58]          txaMonitor.Insert(sMessage)
[59]          Inc $iAttachmentCount
[60]       Endif
[61]    Endif ' IsBody ?
[62]
[63]    Inc $iLevel
[64]      For Each hChild In hPart
[65]          ParsePart(hChild, IsBody) ' Recursive call!
[66]      Next
[67]    Dec $iLevel
[68]
[69]    txaMonitor.Pos = txaMonitor.Length
[70]
[71] End ' ParsePart(...)
[72]
[73] Public Sub OpenMimeMessageFile_Click()
[74]   btnGetStructureMimeMessage.Enabled = False
[75]   Dialog.Title = ("Select a MimeMessage-File ...")
[76]   Dialog.Filter = ["*.txt", ("Text files"), "*", ("All files")]
[77]   Dialog.ShowHidden = False
[78]   Dialog.Path = Application.Path &/ "InBox"
[79]   If Dialog.Openfile(False) Then Return ' False: 'Multiselect' ist ausgeschaltet
[80]
[81]   txaMonitor.Clear
[82]   txaMonitor.Text = File.Load(Dialog.Path)
[83]
[84]   btnGetStructureMimeMessage.Enabled = True
[85]   $iLevel = 0
[86]   $iFlag = 0
[87]   $iAttachmentCount = 0
[88]
[89]   Catch
[90]   Message.Info(Error.Text)
[91]   btnGetStructureMimeMessage.Enabled = True
[92]
[93] End ' OpenMimeMessageFile_Click()
```

Kommentar:

- Der eigentliche Parser wird in der Prozedur *ParsePart(hPart As MimePart, IsBody As Boolean)* beschrieben.
- Sie übergeben dem Parser den EMail-Quelltext als MimeMessage und den Parameter IsBody. Der zweite Parameter ist notwendig, um zwischen einem Text-Teil der MimeMessage (der beim Mime-Typ text/html optional auch (kodierte) Bilder enthalten kann) und den Anhängen zu unterscheiden.
- Beachten Sie den *rekursiven* Aufruf des Parsers in der Zeile 65.

  Minisini schrieb in diesem Zusammenhang: *Moreover, a MIME message is a recursive structure. Usually mail clients hides that by displaying the message contents linearly. So you must parse the parts recursively … .*

- Für eigene Struktur-Analysen benötigen Sie die Quell-Texte von EMails.